

#2

Attorney Docket No. 0826.1801

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Fumirou ABE, et al.

Application No.: TO BE ASSIGNED

Group Art Unit: TO BE ASSIGNED

Filed: HEREWITH

Examiner: TO BE ASSIGNED

For: DATA SORT METHOD, DATA SORT APPARATUS, AND DATA SORT PROGRAM

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN  
APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s)  
herewith a certified copy of the following foreign application:

Japanese Patent Application No. JP2001-227587

Filed: July 27, 2001

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing  
date(s) as evidenced by the certified papers attached hereto, in accordance with the  
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 3/4/02

By: Richard A. Gollhofer  
Richard A. Gollhofer  
Registration No. 31,106

700 11th Street, N.W., Ste. 500  
Washington, D.C. 20001  
(202) 434-1500



JAPAN PATENT OFFICE



This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: July 27, 2001

Application Number: Patent Application  
No. 2001-227587

[ST.10/C]: [JP2001-227587]

Applicant(s): FUJITSU LIMITED

January 11, 2002

Commissioner,

Japan Patent Office Kozo OIKAWA

Certificate No. 2001-3113889

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

JC879 U.S. PTO  
10/086696  
03/04/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office

出 願 年 月 日

Date of Application:

2001年 7月27日

出 願 番 号

Application Number:

特願2001-227587

[ST.10/C ]:

[JP2001-227587]

出 願 人

Applicant(s):

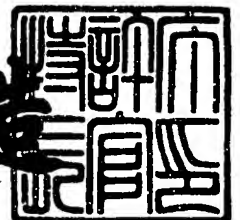
富士通株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2002年 1月11日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3113889

【書類名】 特許願

【整理番号】 0151517

【提出日】 平成13年 7月27日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 7/08

【発明の名称】 データソート方法、データソート装置およびデータソートプログラム

【請求項の数】 5

【発明者】

    【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

    【氏名】 安倍 史郎

【発明者】

    【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

    【氏名】 松浦 正卓

【発明者】

    【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

    【氏名】 田端 洋子

【発明者】

    【住所又は居所】 福岡県福岡市早良区百道浜2丁目2番1号 株式会社富士通九州システムエンジニアリング内

    【氏名】 永田 真彦

【発明者】

    【住所又は居所】 福岡県福岡市早良区百道浜2丁目2番1号 株式会社富士通九州システムエンジニアリング内

    【氏名】 原 泰久

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100072590

【弁理士】

【氏名又は名称】 井桁 貞一

【電話番号】 044-754-2462

【手数料の表示】

【予納台帳番号】 011280

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704486

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データソート方法、データソート装置およびデータソートプログラム

【特許請求の範囲】

【請求項 1】 指定されたソートキー項目に従ってレコードを昇順または降順に並べ替えるデータソート方法であって、

各レコードのソートキー項目の文字列を受理するオートマトンで、かつその最終遷移状態に当該レコード識別子を対応付けたオートマトンを生成するオートマトン作成処理段階と、

前記オートマトンを走査して前記レコードを昇順または降順に順番付けた順序値と当該レコード識別子との組である順序値タプルを生成する順序値タプル生成段階と、

を有することを特徴とするデータソート方法。

【請求項 2】 指定されたソートキー項目に従ってレコードを昇順または降順に並べ替えるデータソート装置であって、

各レコードのソートキー項目の文字列を受理するオートマトンで、かつその最終遷移状態に当該レコード識別子を対応付けたオートマトンを生成するオートマトン作成処理手段と、

前記オートマトンを走査して前記レコードを昇順または降順に順番付けた順序値と当該レコード識別子との組である順序値タプルを生成する順序値タプル生成手段と、

を有することを特徴とするデータソート装置。

【請求項 3】 指定されたソートキー項目に従ってレコードを昇順または降順に並べ替えるデータソートをコンピュータに実行させるためのデータソートプログラムであって、

各レコードのソートキー項目の文字列を受理するオートマトンで、かつその最終遷移状態に当該レコード識別子を対応付けたオートマトンを生成するオートマトン作成処理段階と、

前記オートマトンを走査して前記レコードを昇順または降順に順番付けた順序

値と当該レコード識別子との組である順序値タプルを生成する順序値タプル生成段階と、

をコンピュータに実行させることを特徴とするデータソートプログラム。

【請求項 4】 前記ソートキー項目は複数項目であり、前記順序値タプルは複数の順序値と一つのレコード識別子との組であることを特徴とする請求項 1 記載のデータソート方法。

【請求項 5】 さらに、前記順序値タプルを所定の順に並べたソートテーブルを生成するソートテーブル生成段階を有することを特徴とする請求項 1 または請求項 4 記載のデータソート方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、複数項目からなるレコードが大量に集まった大量レコードデータを、着目する特定の項目をソートキー項目と指定して、各レコードのソートキー項目の値を比較してレコードを順序づけて分類するデータソート技術に係り、特に、CSV形式のデータやXML文書のようなテキスト形式データでレコード構造をもつ文字列データのデータソート技術に関する。

【 0 0 0 2 】

【従来の技術】

従来のデータソートの方法として、選択ソート、挿入ソート、バブルソート、シェルソート、クイックソート等の方法がある。これらの方法はソート単位であるレコードを 1 件ずつ比較してソートを行うため、同一レコードを何度も繰り返して読み出す必要があり、ソート処理に必要な時間はレコード件数の増加に従い級数的に増加する。一般にレコード件数を  $n$  とすると、各方式でのソート時間は以下の数式で表される。

単純選択法： $n^2$

クイックソート： $n \log n \sim n^2$

挿入ソート： $n^2$ （大量件数の場合）

バブルソート： $n \sim n^2$

シェルソート： $n^{1.5} \sim n^{1.25}$

【0003】

【発明が解決しようとする課題】

このように従来のデータソート方法では大量レコードからなるデータをソートするための処理時間はレコード件数の増加に従い級数的に増加するので、データの保守性・運用性を著しく損なう、という課題があった。本発明はこのような課題を解決するために、同一レコードを一度読み出すだけで所望のソート順序値が得られるデータソートの方法、装置およびプログラムを提供することを目的とする。

【0004】

【課題を解決するための手段】

上記課題は、図1に示す如く、各レコードのソートキー項目の文字列を受理するオートマトンで、かつその最終遷移状態に当該レコード識別子を対応付けたオートマトンを生成するオートマトン作成処理段階（ステップS111）と、前記オートマトンを走査して前記レコードを昇順または降順に順番付けた順序値と当該レコード識別子との組である順序値タプルを生成する順序値タプル生成段階（ステップS115）とを有するデータソート方法によって解決される。

【0005】

すなわち、図2のように構成されたデータソート装置の場合、外部から指定されるキー条件kcを入力データdに適用する前段階を経て、ステップS111でオートマトン作成処理手段3は入力レコードを識別するレコード識別子ridをレコード識別子設定手段1から取得し、当該レコードの指定キーの値に対応するキー文字列k-strをキーデータ前処理手段2から取得する。

【0006】

オートマトン作成処理手段3は入力されるすべてのレコードについて、このキー文字列k-strを受理するオートマトンamを生成する。オートマトンamはキー文字列k-strの文字数（可変長の場合はその最大桁数）分の階層をもつ状態遷移表の集合であり、これらの表は相互にリンクされ、図2中に示す如く、初期状態遷移表（00と表示）を根とするツリー構造を形成する。



## 【 0 0 0 7 】

本発明ではこのオートマトン $am$ の各キー文字列 $k-str$  を受理する最終遷移状態を示す最終状態遷移表に当該レコード識別子を対応付けて記録するものである。従って、入力全レコードのレコード識別子 $rid$  がいずれかの状態遷移表に記録されている。

次いで、ソートテーブル作成処理手段 4 中の順序値タプル生成手段41は、ステップ S 1 1 5 において、前記オートマトン $am$ のツリーを根元から順序づけられた状態順に枝にそって深さ優先で走査して得られた状態遷移表に記録されたレコード識別子 $rid$  を取得する。この時、走査順序に従って順序値 $jval$ が当該レコード識別子 $rid$  と対応付けられる。これによって順序値タプル生成手段41は順序値 $jval$ と当該レコード識別子 $rid$  との組である順序値タプル $j-tp1$  を生成する。

## 【 0 0 0 8 】

以上の順序値タプル生成段階は、指定ソートモードが昇順の場合、順序値 $jval$ の最小値から生成されるので、例えば昇順で上位 2 0 のレコードを求める場合、本順序値タプル生成段階を 2 0 ループするだけで所望のレコードが求まる。先のステップ S 1 1 1 のオートマトン作成処理段階で全レコードを一度読み込むだけでこのように所望のソート順序値が得られるので従来技術よりその処理スピードははるかに迅速である。

## 【 0 0 0 9 】

また、全レコードに関するソートを完成させる場合、ステップ S 1 1 5 の順序値タプル生成段階を完遂し、全順序値タプル $j-tp1$  を順序値 $jval$ の昇順または降順に並べたソートテーブルであるソート順ソートテーブル $s-st$ を作成すればよく、やはり従来技術よりその処理スピードははるかに迅速である。

入力キー条件 $kc$ で指定ソートキー項目が複数項目の場合、例えば第 1 優先キー、第 2 優先キー、...、第 $K$  優先キーが設定された場合は、オートマトン $am$ はソートキー項目毎に $K$  個作成される。この場合、1 レコード読み込む毎に、ステップ S 1 1 1 のオートマトン作成処理段階は $K$  回ループし、 $K$  個のオートマトン $am$ を更新成長させる。また、全レコード読み込み終了後、ステップ S 1 1 5 のソートテーブル作成処理段階も $K$  回ループして $K$  個のオートマトン $am$ を順次走査して

、レコード識別子rid の昇順または降順に並べたソートテーブルであるレコード順ソートテーブルr-stを生成する。

#### 【0010】

このように生成したレコード順ソートテーブルr-stでは、各レコード毎にその第1 優先キー順序値jval(1)、第2 優先キー順序値jval(2)、...、第K 優先キー順序値jval(K) の複数順序値タプルが得られる。この複数順序値タプルを文字列とみなして、この文字列を複数順序値キー文字列として、レコード順ソートテーブルr-st中の全タプルに対して、オートマトン作成処理手段3にこれを与え、これを受理するオートマトンを生成し（ステップS111を適用）、このオートマトンをソートテーブル作成処理手段4に与えて走査する（ステップS115を適用）と、前記複数ソートキーに対する総合的な順番として新たに順番付けた総合順序値と当該レコード識別子との組である総合順序値タプルを生成する。従って、ステップS115は総合順序値タプル生成段階となる。ここで得られた総合順序値タプルから総合的なソート順ソートテーブルs-stが得られ、複数キーに対するデータソートが完成する。この場合も全レコードを一度読み込むだけで所望のソート順序値が得られるので、従来技術よりその処理スピードははるかに迅速である。

#### 【0011】

さらに、図1のステップS14に示すように、入力データから読み出したレコード毎に、当該レコードの開始アドレスを当該レコードを識別するレコード識別子と対応付けたタプルを記録するレコード識別子設定段階を設けることで、データd中に明示的なレコード識別子が存在しない場合でもレコード識別子ridを付与でき、本発明の適用範囲が広がる。

#### 【0012】

さらに、図1のステップS19に示すように、指定されたソートキー項目毎のソート方法に関する条件であるキー条件kcに応じて、入力されたソートキー項目データを前記オートマトン作成処理に適するキー文字列k-strに変換するキーデータ前処理段階を設けることで、キーデータのデータ型や桁数をソートに適するように変換をし、ソート可能なデータdの範囲を広げることができる。

## 【 0 0 1 3 】

## 【発明の実施の形態】

本発明のデータソート方法の実施例を図 1 ～図 9 により説明する。なお、本発明におけるコンピュータ処理は、コンピュータプログラムにより当該コンピュータの主記憶装置上で実行されるが、このコンピュータプログラムの提供形態は、当該コンピュータに接続された補助記憶装置をはじめ、フロッピーディスクや C D - R O M 等の可搬型記憶装置やネットワーク接続された他のコンピュータの主記憶装置及び補助記憶装置等の各記録媒体に格納されて提供されるもので、このコンピュータプログラムの実行に際しては、当該コンピュータの主記憶装置上にローディングされ実行されるものである。

## 【 0 0 1 4 】

最初に本発明の中心となるオートマトンの構造例について、図 3 の具体例によって説明する。既に図 2 のツリー図でオートマトン  $am$  を示したように、オートマトン  $am$  は複数の状態遷移表の集合で、キー文字列  $k\text{-str}$  を受理する状態遷移を写像するものである。すなわち、各ノードに状態遷移表を配し、ノード間のリンクをするアローがキー文字列  $k\text{-str}$  の各文字に対応する。一つの状態遷移表から次の状態遷移表へリンクするアローの数が状態遷移数である。オートマトン  $am$  ではキー文字列  $k\text{-str}$  の一文字がとりうる値の種類に応じて状態遷移を行う。よってキー文字列  $k\text{-str}$  の一文字単位のビット列の長さによってオートマトン  $am$  の 1 文字当りの状態遷移数が異なる。その事情を図 3 (a) で説明する。

## 【 0 0 1 5 】

図 3 (a) ではキー項目の値が文字列『富士通』の場合のキー文字列  $k\text{-str}$  の設定例を示した。本例ではこれらの日本語文字は S J I S コードで一文字 2 バイト (16 ビット) で文字数は 3 で、合計 4 8 ビットのビット列である。キー文字列  $k\text{-str}$  の 1 文字単位とみなすビット数 (1 文字単位ビット数) を 1 6 とすると、第 1 文字は  $x 9 5 7 8$  であり、1 文字当り最大遷移状態数は 6 5 5 3 6 である。同様に、1 文字単位ビット数を 8 とすると、第 1 文字は  $x 9 5$  であり、1 文字当り最大遷移状態数は 2 5 6 であり、1 文字単位ビット数を 4 とすると、第 1 文字は  $x 9$  となり、1 文字当り最大遷移状態数は 1 6 である。このように受理するキー

文字列k-str の 1文字単位ビット数によってオートマトンamの構造が異なる。本実施例ではキー文字列k-str の 1文字単位ビット数を4として以下の説明を行うが、本発明のキー項目の文字列を受理するオートマトンamがこれに限定されるものではない。

#### 【0016】

図3(b)にキー文字列k-str を4ビット文字とする第p文字受理状態遷移表tid(p)の構成例を示す。表id欄は本表を識別する識別子であり、同時に本表へのアクセスアドレスxxxxを記録する。文字i受理遷移先欄(i=0,1,...,F)は入力キー文字列k-str の第p文字の内容がiであった場合に遷移する第p+1文字状態遷移表tid(p+1)へのポインタである。図の例では文字0受理遷移先欄にポインタtid(p+1)=yyyyを記入し、第p+1文字状態遷移表tid(p+1)=yyyyにリンクすることを示している。また、文字1受理遷移先はヌル("")であるが、これは本オートマトンamが第p文字1のキー文字列k-strを受理しないことを示す。(すなわち、本発明ではそのようなキー文字列k-strが存在しなかったことを含意する。)

遷移元表id欄は本表に遷移する元の第p-1文字状態遷移表tid(p-1)を示すポインタであり、図の例ではwwwによって示される。なお、オートマトンamのツリーの根は初期状態遷移表tid(0)であるが、当然、初期状態遷移表tid(0)においては遷移元表id欄はヌル("")となっている。

#### 【0017】

最終遷移状態フラグ欄は本表を最終遷移状態とするキー文字列k-strを持つレコードの存在を示すフラグである。本欄の値が0の場合、本表を最終遷移状態とするレコードはない。一方、本実施例では本最終遷移状態フラグを受理件数カウンタとしても利用する。すなわち、最終遷移状態フラグ欄の値はこの表を最終とするキー文字列k-strのレコード件数を示す。図の例では18件が本表を最終遷移状態として受理されている、すなわち、18件の異なるレコードがすべて同一のキー文字列k-strをもつことがわかる。

#### 【0018】

受理レコード識別子リスト欄は上記18件のレコードのレコード識別子ridのリストである。また、受理レコード補助情報欄は上記レコードの補助情報を受理レ

コード識別子リスト欄のリストに対応して持つ。補助情報とは、例えば当該レコードの開始アドレスやレコード長などの関連情報である。これらの情報は図 5 (d) に示すようなレコード識別子設定テーブルrid-t に基づいてとられる。

#### 【 0 0 1 9 】

以上のオートマトンamを利用して、本発明のデータソート装置が動作する。図 2 は本発明のデータソート装置の構成の一例である。図 2 に示す如く、本データソート装置は外部からデータd およびキー条件kcが与えられ、動作を開始する。データd は多数のレコードからなるデータ集合であり、キー条件kcはこのデータd のレコードをソートするキーに関する条件を指定する。図 4 には入力されるデータd の構造とキー条件kc設定例を示す。これに基づいて、本発明の実施例を説明する。

#### 【 0 0 2 0 】

図 4 (a) はデータd の構造例である。図の如く、データd はタグ文字列<R-  
END>によってレコード構造をもつテキスト形式データで、さらにレコード内  
も不等号文字で囲まれたタグ文字列によって項目が識別され区切られている。図  
の例では一つのレコードは部コード、氏名、入社年度、売上の 4 つの項目データ  
からなっている。それぞれのレコードを識別するレコード識別子rid は本例では  
データ中には存在しない。図では最初のレコードをレコード識別子rid = 1、次  
のレコードをレコード識別子rid = 2、... とし、最終レコードはレコード識  
別子rid = Rmaxで合計Rmax件のレコードからなっている。

#### 【 0 0 2 1 】

図 4 (b) にはキー条件kcの設定例を示す。キー条件kcはキー優先順位、キー項  
目名、ソートモード、データ型、最大文字数を設定するものである。キー優先順  
位は複数キーを設定する場合の各キーに与える優先順位である。図の例では 3 つ  
のキー項目<売上><入社年度><部コード>がこの順の優先順位で設定されて  
いる。ソートモードは昇順／降順を指定する。例では<売上>を降順指定し、他  
は昇順としている。

#### 【 0 0 2 2 】

データ型は数値型と文字列型とがある。文字列型は項目データを文字列として

扱うことを指定しており単純である。すなわち、キー文字列k-str は項目データを構成するビット列を4 ビットずつ区切って1 文字と数えるだけでよい。例えば図4 (c) の文字列"-123.456"はこれを文字列と指定するとキー文字列k-str は16文字からなるx2D3132332E343536 となる。

#### 【0023】

これに対し数値型指定は項目データの文字列が数値を表すこと、かつ、ソートに際してはこれを数値として比較することを要求する。従って、キーデータ前処理手段2は該当項目データの文字列が意味する数値を解釈してこれを相互比較可能な標準形文字列であるキー文字列k-str に変換する。数値にはゼロサプレス、小数点、正負符号、仮数指数表現、スペース挿入、全角半角混在などの表現のばらつきがあるので、キーデータ前処理手段2はこれらを解釈して標準の内部文字列に変換する。

#### 【0024】

数値型の内部文字列変換仕様は本実施例では 4バイト (32ビット) の浮動小数点形式であり、符号部 1ビット、指数部は指数符号部 1ビット指数絶対値 7ビットの計8ビット、仮数部は仮数整数値を23ビットでとり、符号部、指数符号部とも負=0、正=1 とする。例えば図4 (c) の文字列"-123.456"はこれを数値型と指定するとキー文字列k-str は8文字からなるx4181E240 となる。これは通常のANSI/IEEE規格754の浮動小数点形式とは異なるが、この4バイト8文字のキー文字列k-str のビット並びが数値としての大小関係を反映するように考慮したものである。

#### 【0025】

また、キー条件kcとして最大文字数を設定すると、当該キー項目データが最大文字数を越える場合に、越えた文字列を無視してソートを行うことを指定する。ここでいう最大文字数はデータとしての文字数でありキー文字列k-str の文字数ではない。以上、キー条件kcの各設定はキー項目名を除いてそれぞれデフォルト設定があり、入力によってこれを変更するものとする。

#### 【0026】

以上でのキー文字列k-str の構造およびそれを受理するオートマトンamの構造

を説明したが、本発明のデータソート装置の動作説明の前に、ソートテーブルについて予備説明を行う。本発明のデータソート装置は後述するように、個々のレコード識別子rid と当該レコードのソート結果の順序を示す順序値jvalとの組である順序値タプルj-tpl を次々につくり出し、これをソートテーブルの形で蓄積していく。図6にソートテーブルのデータ構造例を示すように、本発明ではソート順ソートテーブルs-stとレコード順ソートテーブルr-stの2種類のソートテーブルをもつ。

## 【0027】

図6(a)は順序値タプルj-tplの構造を示す。順序値jvalはソート指定キーが複数の場合にはキー毎に生成されるので複数順序値タプルとも呼ぶ。キー条件kcによって第k優先キーと決められたキーでの順序値jval(k) ( $k=1, 2, \dots, K$ )の組が一つのレコード識別子rid と対応するので図6(a)のような順序値タプルj-tplとなる。これを特定の順序値jvalに注目して順序値jvalの昇順に蓄積したものが図6(b)に示すソート順ソートテーブルs-stである。図に示すように一般に順序値jvalは同一値を複数レコード識別子rid に対応させることができる。すなわち、同一順序(順序をつけられない)のレコードが複数個出現する。

## 【0028】

図6(c)はレコード順ソートテーブルr-stの構造例である。これは図6(a)の複数順序値タプルをレコード識別子rid の値の昇順に蓄積したテーブルである。なお、本実施例では図のように各レコード識別子rid 行にキー毎の順序値jval欄の他にキー欠落フラグ欄を設ける構造とした。キー欠落とは当該レコードの指定キーを探索した場合に指定キー項目が存在しなかった場合、または存在するがその項目データがヌル値であった場合をいう。後述するように、本発明のデータソート装置ではキー欠落を検査して発見すると、このレコード順ソートテーブルr-stの該当キー欠落フラグを立てる。図6(c)ではレコード識別子rid = 3の第1優先キー、レコード識別子rid = 6の第2優先キーなどにキー欠落フラグが立てられている。

## 【0029】

キー欠落時の対処には種々のインプリメントがあるが、本実施例ではそのよう

なレコードは該当キーの順序値  $jval$  が最大であるとする。すなわち、昇順、降順にかかわらずこれらのレコードは最下位にソートされるものとした。図 6 (c) ではこの最下位の順序値  $jval$  を  $\max(k)$  と表しているが、その値は後述するように順序値タプル生成段階で決定される。

#### 【 0 0 3 0 】

以上の準備を踏まえて、本発明のデータソート装置の動作を図 1、5、7、8、9 の動作フローに従って説明する。図 1 は本発明のデータソート方法の動作フロー例である。ステップ S 1 1 は初期設定段階であり、図 5 (a) にその詳細動作を示す。

まず、レコード識別子設定手段 1 は図 5 (d) に示すレコード識別子設定テーブル  $rid-t$  の領域設定を行う。次いでレコード識別子設定手段 1 は既読レコード数変数  $rr$  をリセットする。( $rr \leftarrow 0$ )。本既読レコード数変数  $rr$  は、後述のレコード識別子設定段階 (ステップ S 1 4) でレコード識別子設定手段 1 が各レコード読み込み時に対応するレコード識別子  $rid$  を発行する拠り所となるものである。キーデータ前処理手段 2 はキー条件  $kc$  を読み込み格納する。次いで、レコード識別子設定手段 1 はキーデータ前処理手段 2 からキー項目数を得て、先に図 6 で説明したレコード順ソートテーブル  $r-st$  の領域設定を行い、また、ソート順ソートテーブル  $s-st$  の領域設定を行う。ここでは、単にこれらの領域を確保するのみでその値はすべてヌルクリアされている。次いでレコード識別子設定手段 1 は第  $k$  優先キー初期状態遷移表  $tid-k(0)$  ( $k=1, 2, \dots, K$ ) を作成する。図 4 (b) のように 3 個のキーが設定された場合は  $K=3$  で、第 1 優先キー初期状態遷移表  $tid-1(0) = ii\ ii(1)$ 、第 2 優先キー初期状態遷移表  $tid-2(0) = iii\ ii(2)$ 、第 3 優先キー初期状態遷移表  $tid-3(0) = iiii\ (3)$  が生成される。 $iiii(k)$  は各初期状態遷移表  $tid-k(0)$  のアドレスでかつ表  $i\ d$  である。

#### 【 0 0 3 1 】

次いで、データソート装置の共通制御部 (図 2 には図示せず) は図 1 のステップ S 1 2 でデータ  $d$  の 1 レコードを読み込む。ステップ S 1 3 でファイル終了でなければステップ S 1 4 でレコード識別子設定段階の動作を行う。ここでは図 5 (b) に示すように、まず、既読レコード数変数  $rr$  をインクリメントし ( $rr \leftarrow [rr]$ )



+1)、これを図 5 (d) のレコード識別子設定テーブルrid-t のレコード識別子rid 欄に設定し (レコード識別子rid  $\leftarrow$  [rr])、開始アドレスオフセット値、レコード長の設定を行う。なお、このレコード識別子設定テーブルrid-t は、レコード識別子rid によって該当レコードをダイレクトアクセスするためのインデクスとして使用されるものである。

#### 【 0 0 3 2 】

これによって当該レコードに対するレコード識別子rid の設定ができたので、次いで、先に領域設定されたレコード順ソートテーブルr-stの所定行にこのレコード識別子rid  $\leftarrow$  [rr] を登録する。

次いで、ステップ S 1 5、1 6、1 7 は第k 優先キー ( $k=1, 2, \dots, K$ ) の処理に移行するループ制御である。ステップ S 1 8 ではキーデータ前処理手段 2 は当該レコードに当該第k 優先キーが存在するか、またそのキーデータがヌル値でないか进行检查し、キー欠落でなければステップ S 1 9 でキーデータ前処理を行う。この処理は先に図 4 (c) で述べたように、当該第k 優先キーのキー条件kcに基づいて、キーデータをキー文字列k-str に変換するものである。またステップ S 1 8 でキー欠落と判定の場合はステップ S 1 1 0 に移行し図 5 (c) に示すキー欠落処理段階の動作を行う。

#### 【 0 0 3 3 】

図 5 (c) によってこれを示すと、まず、キーデータ前処理手段 2 は図 6 (c) のレコード順ソートテーブルr-stの当該レコード識別子rid の行の第k 優先キー欠落フラグを立てる。次いで、第k 優先キー初期状態遷移表tid-k(0)の最終遷移状態フラグ欄を1 加算し、受理レコード識別子リスト欄に当該レコード識別子rid をアペンドする。すなわち、キー欠落レコードは当該キーのオートマトンamの初期状態遷移表tid-k(0)の受理レコード識別子リスト欄に記入することによって分類される。

#### 【 0 0 3 4 】

図 1 に戻り、正常にキー文字列k-str が取得できた場合はステップ S 1 1 1 に達し、ここでオートマトン作成処理手段 3 は図 7 に示す第k 優先キー対応のオートマトンamの更新を行う。図 7 には現在のオートマトンamが新たにキー文字列k-

str = c(1), c(2), ..., c(C) を受理するように更新される動作例を示す。

まずステップ S 7 1 でオートマトン作成処理手段 3 は第 k 優先キーのキー文字列 k-str の文字を取り出す文字ポインタ cp をリセットし、第 k 優先キーのオートマトン am から状態遷移表を取り出す表ポインタ tp をオートマトン am の根（すなわち、初期状態遷移表 tid-k(0)）にポイントする。

#### 【0035】

次いでステップ S 7 2、7 3 によって文字ポインタ cp を一つ進め最終文字に達するまでループ制御する。第 cp 文字 c(cp) が存在するときはステップ S 7 4 に、全文字終了時はステップ S 7 9 に分岐する。

ステップ S 7 4 ではオートマトン作成処理手段 3 は当該文字 c(cp) のコード [c(cp)] = i を調べ、当該状態遷移表 tp の文字 i 受理遷移先欄がヌルかを検査する。ヌルであった場合はまだかつて受理されていないキー文字列 k-str であるのでステップ S 7 5 で対応する新たな状態遷移表を生成する。すなわち、オートマトン作成処理手段 3 は新しい状態遷移表生成アドレス = aaaa を取得し、第 cp 文字状態遷移表 tid-k(cp) を生成し、その表 i d 欄に tid-k(cp) = aaaa を記入する。また、その遷移元表 i d 欄に元の状態遷移表ポインタ値 [tp] を記入する。また、これと対応して、ステップ S 7 6 で遷移元表 (表 i d [tp] = tid-k([cp]-1) の表) の文字 [c(cp)] 受理遷移先欄に値 tid-k(cp) = aaaa を記入する。これはステップ S 7 4 で検査してヌルであった [tp {文字 [c(cp)] 受理遷移先}] を値 aaaa と書き換えることに他ならない。

#### 【0036】

かくして、キー文字列 k-str の第 c(cp) 文字まで受理したので、次の文字を検査すべくステップ S 7 7 において、表ポインタ tp をステップ S 7 5 で生成した状態遷移表へと進める。一方、ステップ S 7 4 で [tp {文字 [c(cp)] 受理遷移先}] がヌルでなかった場合は既にこの文字を受理した状態遷移表が存在し、その状態遷移表のアドレスが得られているので、オートマトン作成処理手段 3 はステップ S 7 8 でこれを新たに表ポインタ tp として設定する。

#### 【0037】

ステップ S 7 7 またはステップ S 7 8 で表ポインタ tp を次文字用に設定し、ス

テップ S 7 2 に戻り文字ポインタ cp を一つ進めて次の文字に対処する。全文字終了時にステップ S 7 9 に到達する。この時点での表ポインタ tp の値 tid-k(C) で示す状態遷移表は当該キー文字列 k-str を受理する最終遷移状態を示している。よってオートマトン作成処理手段 3 はステップ S 7 9 でこの表 tp の最終遷移状態フラグ欄を 1 加算し、受理レコード識別子リスト欄に値 [rr] (すなわち、当該受理キー文字列 k-str をキーデータとするレコードのレコード識別子 rid) をアペンドする。また受理レコード補助情報欄に関連設定を行う。

## 【 0 0 3 8 】

図 1 に戻り、ステップ S 1 1 1 の第 k 優先キー対応オートマトン作成処理段階を終えるとステップ S 1 6 に戻り次の第 k+1 優先キーに対応するオートマトン am について同様処理を行う。かくして当該 1 レコード中の複数キーの全部に対して K 個のオートマトン am の作成更新が終了すると、ステップ S 1 7 でループを外れ、ステップ S 1 2 に戻りデータソート装置の共通制御部はデータ d の次の 1 レコードを読み込む。

## 【 0 0 3 9 】

このように、すべてのレコードを読み、これに対応するレコード識別子設定、オートマトン am 更新を終え、ファイル終了を検出すると、データソート装置の共通制御部はステップ S 1 1 2 に移り、これまで生成更新したオートマトン am を走査する段階に至る。ステップ S 1 1 2 ~ 1 1 4 は第 k 優先キー毎のループ制御を行い、ステップ S 1 1 5 でソートテーブル作成処理手段 4 は当該第 k 優先キー対応の順序値タプル生成およびソートテーブル生成を行う。その動作を図 8 によって以下に説明する。

## 【 0 0 4 0 】

図 8 のステップ S 8 1 は順序値タプル生成の初期設定で、先に説明したキー欠落レコードを同定する。すなわち、ソートテーブル作成処理手段 4 の順序値タプル生成手段 41 は表ポインタ tp を第 k 優先キーオートマトン am の初期状態遷移表 tid-k(0)=iiii(k) に設定し ( $tp \leftarrow \text{tid-k}(0)$ )、コードインデックス code を 0 とする ( $\text{code} \leftarrow 0$ )。 (コードインデックス code については後述する。) そして初期状態遷移表 tid-k(0)=iiii(k) の最終遷移状態フラグ欄の値を同位値カウンタ scount に

設定する（同位値カウンタ  $scount \leftarrow [tp \text{ \{最終遷移状態フラグ\}}]$ ）。これによって第  $k$  優先キー欠落レコードが  $[scount]$  件あることがわかる。一方、既読レコード数変数  $rr$  の値は全レコード件数  $Rmax$  を示しているはずである。よって  $[rr] - [scount] = Rmax - [scount]$  件のレコードを走査対象とするのであるから、すべてのキー欠落レコードの順序値  $jval$  は最下位値  $Rmax - [scount] + 1$  が与えられるべきである。

#### 【 0 0 4 1 】

順序値タプル生成手段41は順序値タプル  $j\text{-tpl}$  を生成するために、レコード識別子ワーク  $ridwk$  および順序値ワーク  $jvalwk$  からなる領域順序値タプルワーク  $j\text{-tplwk}$  を用いる。ステップ S 8 1 では上記のキー欠落レコードの順序値  $jval$  である  $Rmax - [scount] + 1$  を順序値ワーク  $jvalwk$  に設定する（図 8 ステップ S 8 1 の  $jvalwk = \max(k) \leftarrow Rmax - [scount] + 1$ ）。同時に、この値が先に図 6 (c) で説明した  $\max(k)$  でもあるので、ステップ S 8 2 ～ 8 4 において順序値タプル生成手段41はレコード順ソートテーブル  $r\text{-st}$  上およびソート順ソートテーブル  $s\text{-st}$  上の該当位置にもこの値を設定する。ステップ S 8 3 では初期状態遷移表  $tid\text{-}k(0) = iiii(k)$  の受理レコード識別子リスト欄のリスト要素（ $[scount]$  件ある）を順に取り出してレコード識別子ワーク  $ridwk$  に代入して順序値タプルワーク  $j\text{-tplwk}$  を完成している。

#### 【 0 0 4 2 】

以上によってキー欠落レコードに対する順序値  $jval$  の設定が完了したので、順序値タプル生成手段41は次にステップ S 8 5 以降で第  $k$  優先キーオートマトン  $am$  を走査して、各レコード識別子  $rid$  に第  $k$  優先キーの順序値  $jval$  を対応付ける。このとき、順序値タプル生成手段41がステップ S 8 5 以降でとる戦略は以下のようなものである。すなわち、まず、順序値  $jval$  を与えるインデックス変数として順序値インデックス  $jind$  を使用する。最初に順序値インデックス  $jind$  を初期値に設定しておいて、オートマトン  $am$  の第  $k$  優先キー初期状態遷移表  $tid\text{-}k(0)$  から出発してすべてのリンクする状態遷移表を走査する。このときアクセスする状態遷移表を一つ進める毎に順序値インデックス  $jind$  をソートモードの昇順／降順に応じて調整設定し、該当の状態遷移表の受理レコード識別子リスト欄のレコード識

別子rid にその順序値インデックスjind値を順序値jvalとして対応付けるのである。この戦略から明らかなように、順序値タプル生成手段41の状態遷移表アクセス順序が第k 優先キー文字列k-str のソート順序に対応するようなアクセス順序でなければならない。

#### 【0043】

キー文字列k-str の昇順で状態遷移表をアクセスする順序について考えると、まず、第1文字c(1)の値が最小すなわちコード0 のキー文字列k-str に対応する状態遷移表を選ぶ。これは第k 優先キー初期状態遷移表tid-k(0)の文字0 受理遷移先欄のポインタが示す第1 文字状態遷移表tid-k(1)である。もしもこの第1 文字状態遷移表tid-k(1)の最終遷移状態フラグ欄が1 以上であるなら、それはキー文字列k-str がコード0 の1 文字であるレコードがあることを意味する。(先に説明したようにここでいう文字はキー文字列k-str の文字単位、本実施例では4ビット、であり、実際の文字コード体系での文字の割りあての有無についてはここでは問わない。)このようにして最上位の順序値jvalがこの第1 文字状態遷移表tid-k(1)の受理レコード識別子リスト欄に記入されたリスト要素の各レコード識別子rid に対応付けられる。その件数分順序値インデックスjind値を増やして次の順序値jvalを用意して、次はこの第1 文字状態遷移表tid-k(1)の文字0 受理遷移先欄に書かれた第2 文字状態遷移表tid-k(2)にアクセスする。

#### 【0044】

このように順序値タプル生成手段41はオートマトンamのツリー構造を深さ優先でたどり、また、文字0 受理遷移先欄から文字F 受理遷移先欄の最大16個の枝についてはこの順序で走査を進める。状態遷移表を深さ優先でたどるために表ポインタtpを用いるが、分岐枝を文字0 受理遷移先欄から文字F 受理遷移先欄へたどるためにステップS81で設定したコードインデックスcodeを使用する。あるノードtpの状態遷移表の分岐枝code(すなわち文字[code]受理遷移先欄)から出発して深さ優先の走査処理を終えると、またこのノードtpの状態遷移表の分岐枝codeに戻って次の枝に走査を移すバックトラック制御が必要である。これを実現するために、順序値タプル生成手段41は走査のバックトラック用に、表ポインタtpとコードインデックスcodeとの組をスタックするバックトラック用スタックを用

いる。本スタックはlast-in-first-out 型のスタックであって、ポップ命令によって、最後にプッシュされた値が読み出される構造である。

#### 【 0 0 4 5 】

以上の戦略を実現する動作フローを図 8 のステップ S 8 5 以降で示す。ステップ S 8 5 では、順序値タプル生成手段 41 は順序値 jval を与える順序値インデックス jind を初期設定するが、その値は当該第 k 優先キーのソートモードの昇順／降順によって異なる。既に述べた戦略で走査すると昇順走査であるのでソートモードが昇順の場合には順序値インデックス jind は初期値 0 からインクリメントするが、ソートモードが降順の場合は順序値インデックス jind は最下位値  $[\max(k)]$  を初期値としてデクリメントする。

#### 【 0 0 4 6 】

先にステップ S 8 1 で表ポインタ tp は第 k 優先キー初期状態遷移表 tid-k(0) を指し、コードインデックス code は 0 を指しているのを、はじめてステップ S 8 6 にはいった時点ではオートマトン am ツリーのルートノードの文字 [code] 受理遷移先欄を出発点とする。一般には第 cp-1 文字状態遷移表 tid-k(cp-1) ノードの文字 [code] 受理遷移先欄を指定してステップ S 8 6 に到達する。コードインデックス code が x10 に達していない場合、すなわち、コードインデックス code が値 0 ～ F である場合、ステップ S 8 7 で当該状態遷移表の文字 [code] 受理遷移先欄に次リンク先があるかを検査し、ない場合はステップ S 8 8 でコードインデックス code をインクリメントする。ステップ S 8 7 でリンク先があった場合、順序値タプル生成手段 41 はステップ S 8 9 でまず現在の表ポインタ tp、コードインデックス code をバックトラック用スタックにプッシュし、ステップ S 8 10 で該当のリンク先すなわち、[tp {文字 [code] 受理遷移先}] を新たに表ポインタ tp に設定してステップ S 8 11 の順序値タプル生成・ソートテーブル作成ルーチンに分岐する。このルーチンについては図 9 で後述する。

#### 【 0 0 4 7 】

ステップ S 8 11 で目的の順序値タプル j-tp1 やソートテーブルの作成更新を終えたと、その状態遷移表について、分岐枝をたどるべくステップ S 8 8 でコードインデックス code をインクリメントし再びステップ S 8 6 の未処理枝有無判定

に入る。コードインデックスcodeがx10 に達した場合はこれ以上分岐枝がないので当該状態遷移表での処理終了となりステップS 8 1 2でバックトラック用スタックから次の分岐先をポップする。これによって最後にプッシュされた表ポインタtpとコードインデックスcodeの位置を取り出すことができる。

#### 【0 0 4 8】

ステップS 8 1 3はポップされた表ポインタtpがないこと、すなわち最終スタックを処理し終えて、全バックトラックが終了したことを検査して、y e s ならば全処理終了となる。ポップされた表ポインタtpとコードインデックスcodeが存在する時は、順序値タプル生成手段41はステップS 8 1 4でこれをもとにこの状態遷移表の次の分岐枝に進む、すなわち、ステップS 8 8に戻り処理を継続する。

#### 【0 0 4 9】

最後にステップS 8 1 1において順序値タプルj-tpl を取得し、ソートテーブルを更新する動作を図9によって説明する。本順序値タプルj-tpl 生成・ソートテーブル生成ルーチンは入力として、順序値インデックスjind 、ソートモードフラグsm(k) および表ポインタtpを与えてコールし、出力として順序値タプルj-tpl ( 順序値タプルワークj-tplwk として与える) の他、図6(c) に示すレコード順ソートテーブルr-stの該当レコード識別子rid 行を更新する。また、必要に応じて図6(b) のソート順ソートテーブルs-stを生成する。例えばK=1 すなわちキーの数が1 個の場合はソート順ソートテーブルs-stを作成すれば目的のソートを実行したことになる。この際にはレコード順ソートテーブルr-stの作成を省くことも可能である。また、本ルーチンは入力の順序値インデックスjindを用いて所定の順序値jvalを発行し、最後に順序値インデックスjindを更新して出力し、次の走査に供する。

#### 【0 0 5 0】

図9のステップS 9 1は指定された状態遷移表tpの最終遷移状態フラグ欄を参照して順序値jvalを発行すべきレコード識別子rid の件数を知り、これを同位値カウンタscountにセットする。そしてソートモードフラグsm(k) が昇順の場合は順序値インデックスjind値+1から順序値jvalを発行すべきなので、順序値ワーク

jvalwkに順序値インデックスjind値+1をセットする。一方、ソートモードフラグsm(k) が降順の場合は、[scount] 件分に対して同一の順序値jvalを発行してその次の下位が現在の順序値インデックスjind値になるべきであるので、発行すべき順序値jvalの値は[jind]-[scount] となり、順序値ワークjvalwkには[jind]-[scount] をセットする。

#### 【 0 0 5 1 】

次いで順序値タプル生成手段41はステップS 9 2～9 4で[scount] 件分のループ制御を行い、ステップS 9 5で順序値ワークjvalwkに対応すべきレコード識別子ワークridwk を取得する。すなわち、状態遷移表tpの受理レコード識別子リスト欄には対応すべき[scount] 件分のレコード識別子rid を要素とするリストが書かれてあるので、これらすべてのレコード識別子rid に同一の順序値ワークjvalwkが対応する。かくしてレコード識別子ワークridwk に該当のレコード識別子rid を設定して順序値タプルワークj-tplwk (ridwk,jvalwk)を生成する。同時に、レコード順ソートテーブルr-st上の該当するレコード識別子rid 行の第k 優先キー順序値jval列のセルに順序値ワークjvalwkの値を転記する。これによってレコード順ソートテーブルr-stの更新が行われた。また、K=1 すなわち、ソートキーが1 個の場合は、上記レコード順ソートテーブルr-stに加え、ソート順ソートテーブルs-st上にも(ridwk,jvalwk)を転記することにより目的のソートを実行したことになる。すなわち、キーが一つで昇順指定であれば、図8の順序値タプル生成動作実行に従い、ソート順ソートテーブルs-stは順序値jvalの上位順に埋まって行くので、オートマトンamの全状態遷移表を走査し終えなくとも、上位20位までのレコード識別子rid を得る、などの用途では、上記ソート順ソートテーブルs-stの更新中途段階で十分ソートの目的が達成されたことになるので、従来技術のデータソートに比べ一層高速のレスポンスを得ることになる。

#### 【 0 0 5 2 】

以上で順序値タプル生成、ソートテーブル作成更新を終えると、最後に次の処理に備えてステップ9 6で順序値インデックスjindを更新する。[scount] 件分の処理を行ったので、ソートモードフラグsm(k) が昇順の場合は順序値インデックスjind値は[jind]+[scount] に、降順の場合は[jind]-[scount] に更新する。



以上、本実施例ではK 個のソートキーが指定された場合に図 6 (c) のレコード順ソートテーブルr-stが形成される動作について述べた。各レコード識別子rid に対応して第1 優先キー順序値jvalから第K 優先キー順序値jvalまでのすべてのセルが順序値jvalで埋められている。本レコード順ソートテーブルr-stから総合的な順序値、すなわち、第1 優先キー順序値jvalが等しいレコード識別子rid 間では第2 優先キー順序値jvalの昇順に、第1 ～第2 優先キー順序値jvalまでが等しいレコード識別子rid 間では第3 優先キー順序値jvalの昇順に、...,第1 ～第K-1 優先キー順序値jvalまでが等しいレコード識別子rid 間では第K 優先キー順序値jvalの昇順に全レコード識別子rid を順序づける総合順序値を与える必要がある。この総合順序値と当該レコード識別子との組である総合順序値タプルを生成する総合順序値タプル生成段階について以下に述べる。

#### 【0 0 5 3】

本段階は図 2 のデータソート装置において、データd として、上述の作成されたレコード順ソートテーブルr-stそのものを与えることにより行われる。すなわち、図 6 (c) のレコード順ソートテーブルr-stの各レコード識別子rid は( 第1 優先キー順序値jval, 第2 優先キー順序値jval,..., 第K 優先キー順序値jval) の複数順序値の列をもっているので、この複数順序値の列を一つの文字列に置き換えて、これを当該レコード識別子に属する文字列とみなし、この文字列を複数順序値キー文字列fjk として前記レコード順ソートテーブル中の全タプルに対して、図 2 のデータソート装置に与えて動作させる。この複数順序値キー文字列fjk の昇順に総合順序値が順序付けられる。従って、本データソート装置の動作でキー数1 でキー項目が複数順序値キー文字列fjk の場合のステップS 9 5 の動作で得られたソート順ソートテーブルs-stは総合順序値順にレコード識別子rid を並べたテーブルとなり所期の結果が得られる。

#### 【0 0 5 4】

(付記 1) 指定されたソートキー項目に従ってレコードを昇順または降順に並べ替えるデータソート方法であって、

各レコードのソートキー項目の文字列を受理するオートマトンで、かつその最終遷移状態に当該レコード識別子に対応付けたオートマトンを生成するオートマ

トン作成処理段階と、

前記オートマトンを走査して前記レコードを昇順または降順に順番付けた順序値と当該レコード識別子との組である順序値タプルを生成する順序値タプル生成段階と、

を有することを特徴とするデータソート方法。

【 0 0 5 5 】

(付記 2) 指定されたソートキー項目に従ってレコードを昇順または降順に並べ替えるデータソート装置であって、

各レコードのソートキー項目の文字列を受理するオートマトンで、かつその最終遷移状態に当該レコード識別子を対応付けたオートマトンを生成するオートマトン作成処理手段と、

前記オートマトンを走査して前記レコードを昇順または降順に順番付けた順序値と当該レコード識別子との組である順序値タプルを生成する順序値タプル生成手段と、

を有することを特徴とするデータソート装置。

【 0 0 5 6 】

(付記 3) 指定されたソートキー項目に従ってレコードを昇順または降順に並べ替えるデータソートをコンピュータに実行させるためのデータソートプログラムであって、

各レコードのソートキー項目の文字列を受理するオートマトンで、かつその最終遷移状態に当該レコード識別子を対応付けたオートマトンを生成するオートマトン作成処理段階と、

前記オートマトンを走査して前記レコードを昇順または降順に順番付けた順序値と当該レコード識別子との組である順序値タプルを生成する順序値タプル生成段階と、

をコンピュータに実行させることを特徴とするデータソートプログラム。

【 0 0 5 7 】

(付記 4) 付記 3 記載のプログラムを記録したコンピュータ読み取り可能な記録媒体。

(付記 5) 前記ソートキー項目は複数項目であり、前記順序値タプルは複数の順序値と一つのレコード識別子との組であることを特徴とする付記 1 記載のデータソート方法。

## 【 0 0 5 8 】

(付記 6) さらに、前記順序値タプルを所定の順に並べたソートテーブルを生成するソートテーブル生成段階を有することを特徴とする付記 1 または付記 5 記載のデータソート方法。

(付記 7) 前記ソートテーブル生成段階において、順序値の昇順または降順に並べたソートテーブルであるソート順ソートテーブルを生成することを特徴とする付記 6 記載のデータソート方法。

## 【 0 0 5 9 】

(付記 8) 前記ソートテーブル生成段階において、レコード識別子の昇順または降順に並べたソートテーブルであるレコード順ソートテーブルを生成することを特徴とする付記 6 記載のデータソート方法。

(付記 9) ソートキー項目は複数項目であり、複数の順序値と一つのレコード識別子との組である複数順序値タプルに基づいて、レコード順ソートテーブルを生成する付記 8 記載のデータソート方法であって、さらに、

前記レコード順ソートテーブルの複数順序値の列を当該レコード識別子に属する文字列とみなし、この文字列を複数順序値キー文字列として前記レコード順ソートテーブル中の全タプルに対してこれを受理するオートマトンを生成するオートマトン作成処理段階と、

前記オートマトンを走査して、前記複数ソートキーに対する総合的な順番として新たに順番付けた総合順序値と当該レコード識別子との組である総合順序値タプルを生成する総合順序値タプル生成段階と、

を有することを特徴とする付記 8 記載のデータソート方法。

## 【 0 0 6 0 】

(付記 1 0) 入力データから読み出したレコード毎に、当該レコードの開始アドレスを当該レコードを識別するレコード識別子と対応付けたタプルを記録するレコード識別子設定段階、をさらに有することを特徴とする付記 1 記載のデ

ータソート方法。

(付記 1 1) 指定されたソートキー項目毎のソート方法に関する条件であるキー条件に応じて、入力されたソートキー項目データを前記オートマトン作成処理に適するキー文字列に変換するキーデータ前処理段階、をさらに有することを特徴とする付記 1 記載のデータソート方法。

【 0 0 6 1 】

【発明の効果】

以上の説明から明らかなように、本発明によれば、各レコードのソートキー項目の文字列を受理するオートマトンを生成し、その最終遷移状態に当該レコード識別子を対応付け、このオートマトンを走査することにより、従来のデータソート方法より高速なデータソートが可能となる、という効果がある。

【図面の簡単な説明】

【図 1】 本発明のデータソート方法の動作フロー例

【図 2】 本発明のデータソート装置の機能ブロック図

【図 3】 状態遷移表によるオートマトン生成例

【図 4】 入力されるデータ d の構造とキー条件 kc 設定例

【図 5】 オートマトン am 生成準備の各段階の動作例とレコード識別子設定テーブル rid-t

【図 6】 ソートテーブルのデータ構造例

【図 7】 オートマトン作成処理段階の動作フロー例

【図 8】 ソートテーブル作成処理段階の動作フロー例

【図 9】 順序値タプル j-tp1 生成・ソートテーブル生成ルーチンの動作例

【符号の説明】

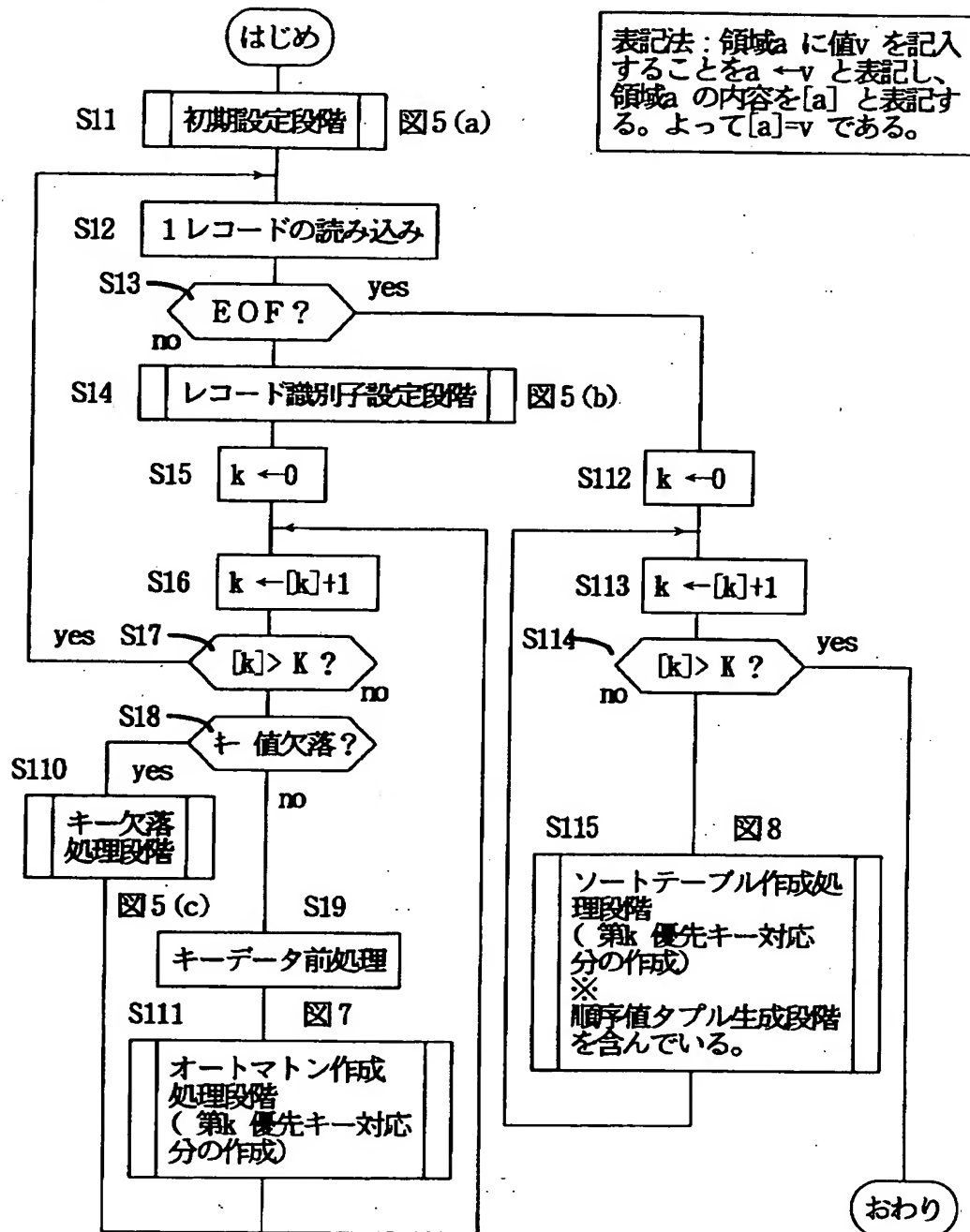
- 1 レコード識別子設定手段
- 2 キーデータ前処理手段
- 3 オートマトン作成処理手段
- 4 ソートテーブル作成処理手段
- 41 順序値タプル生成手段
- k-str キー文字列

kc キー条件  
 am オートマトン  
 tid(0) 初期状態遷移表  
 tid(p) 第p 文字状態遷移表  
 j-tpl 順序値タプル  
 rid レコード識別子  
 jval 順序値  
 s-st ソート順ソートテーブル  
 r-st レコード順ソートテーブル  
 rid-t レコード識別子設定テーブル  
 tp 表ポインタ  
 cp 文字ポインタ  
 scount 同位値カウンタ  
 code コードインデックス  
 rr 既読レコード数変数

【書類名】 図面

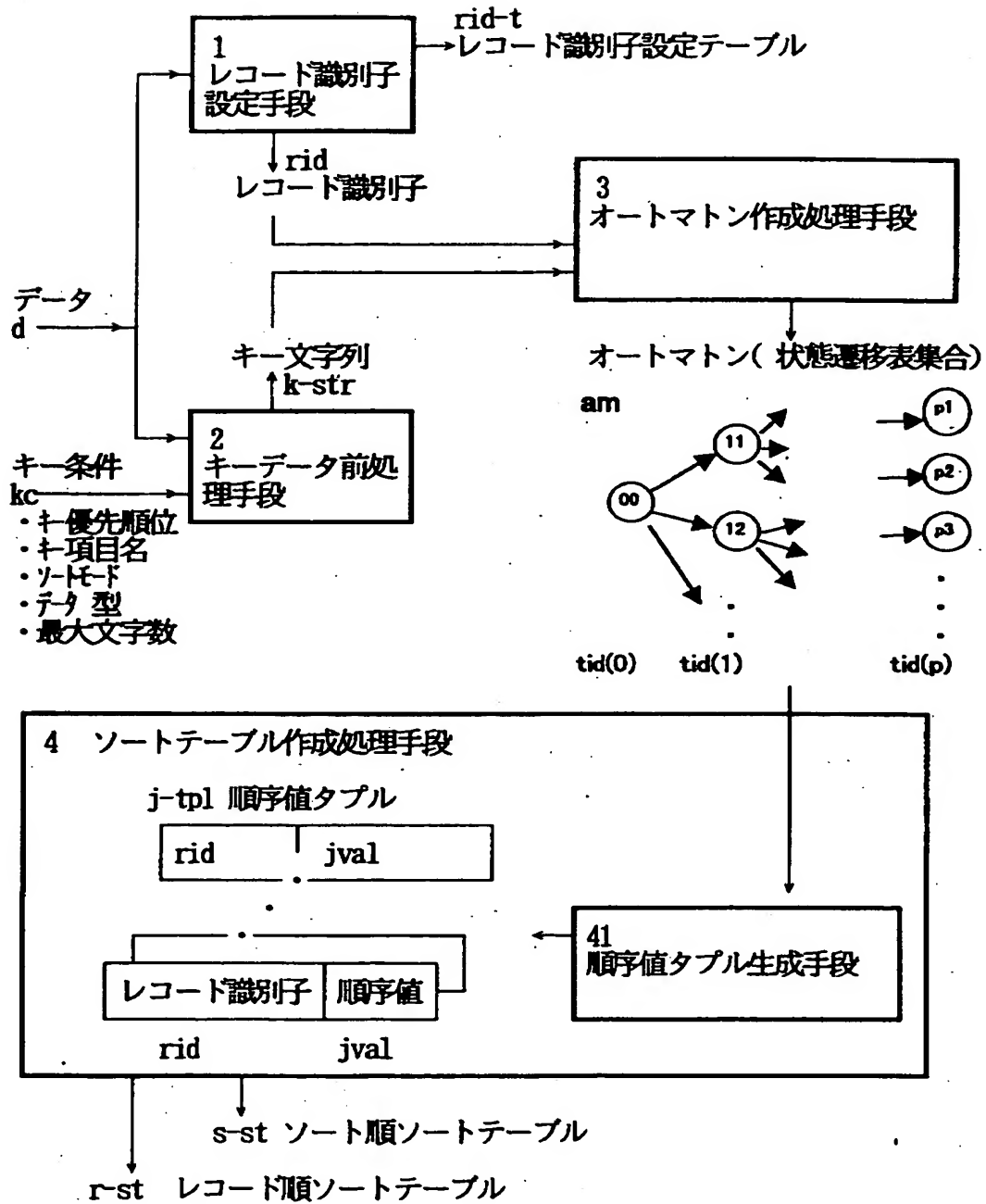
【図 1】

本発明のデータソート方法の動作フロー例



【図 2】

本発明のデータソート装置の機能ブロック図



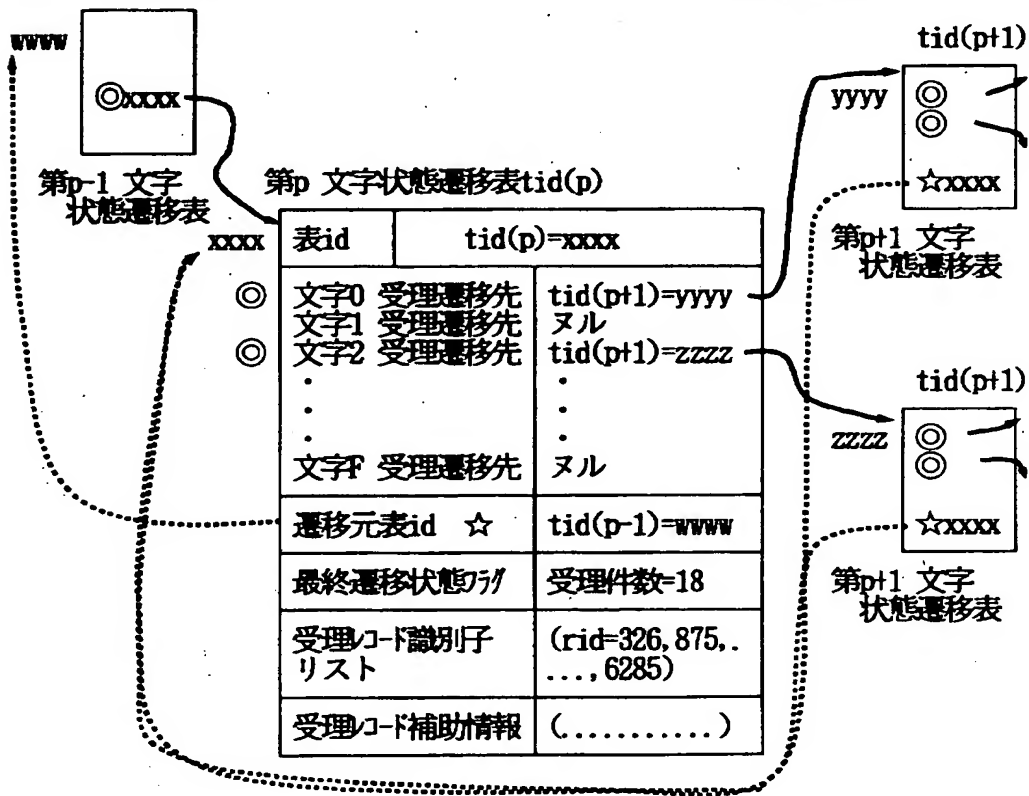
【図 3】

## 状態遷移表によるオートマトン生成例

(a) キー項目の値が文字列「富士通」の場合のキー文字列k-str の設定例

	文字単位ビット数	16	8	4
文字「富」のSJISコード 9578	キー文字列k-str 第1文字	9578	95	9
	第2文字	8E6D	78	5
	第3文字	92CF	8E	7
文字「土」のSJISコード 8E6D	第4文字	・	6D	8
	第5文字	・	92	8
	第6文字	・	CA	E
	・	・	・	・
文字「通」のSJISコード 92CF	・	・	・	・
	1文字当り最大遷移状態数	65536	256	16

(b) キー文字列k-str を4ビット文字とする第p文字受理状態遷移表の構成例





【図 4】

## 入力されるデータd の構造とキー条件kc設定例

## (a) データd の構造例

```

ワード識別子rid =1 → <部コード>15<氏名>安倍太郎<入社年度>90<売上>900<R-END>
ワード識別子rid =2 → <部コード>01<氏名>松浦一郎<入社年度>90<売上>900<R-END>
      .               <部コード>15<氏名>田嶋花子<入社年度>92<売上>605<R-END>
      .               <部コード>07<氏名>永田正夫<入社年度>95<売上>850<R-END>
      .               <部コード>02<氏名>原裕太<入社年度>97<売上>605<R-END>
ワード識別子rid =Rmax      . . . . . <R-END>      EOF

```

## (b) キー条件kc設定例

キー優先順位	1	2	3
キー項目名	<売上>	<入社年度>	<部コード>
ソートモード	降順	昇順	昇順
データ型	型=数値 ゼロサプレス=あり 小数点=あり 正負符号=あり 仮数指数表現=あり スペース挿入=あり 全角半角混在=あり	型=数値 ゼロサプレス=なし 小数点=なし 正負符号=なし 仮数指数表現=なし スペース挿入=なし 全角半角混在=なし	型=文字列
最大文字数	10	—	—

## (c) キー条件kcに基づくキーデータ前処理の例

キー条件kc	キーデータ	入力文字列	キー文字列k-str	文字数
型=文字列	-123.456	x2D3132332E343536	x2D3132332E343536	16
型=数値	2	x32	x00000002	8
	15	x3135	x0800000F	8
	03	x3033	x00000003	8
	-123.456	x2D3132332E343536	x4181E240	8

数値型の内部文字列変換仕様：浮動小数点形式

符号部 1ビット 負=0、正=1 (※)

指数部 指数符号部 1ビット 負=0、正=1 (※)

指数絶対値 7ビット

仮数部 仮数整数値 23ビット

(※) は通常のANSI/IEEE規格754浮動小数点形式と異なる。

【図 5】

オートマトン $\alpha$ 生成準備の各段階の動作例とレコード識別子設定テーブル $rid-t$

(a) 初期設定段階の動作例 (図1ステップS11)

レコード識別子設定手段1はレコード識別子設定テーブル $rid-t$ の領域設定を行う。また、既読レコード数変数 $rr$ をリセットする。 $(rr \leftarrow 0)$   
 キーデータ前処理手段2はキー条件 $kc$ を読み込み格納する。  
 キーデータ前処理手段2からキー項目数を得て、レコード順ソートテーブル $r-st$ の領域設定を行う。  
 ソート順ソートテーブル $s-st$ の領域設定を行う。  
 第 $k$  優先キー初期状態遷移表 $tid-k(0)$ を作成する。 $tid-k(0) = iiii(k)$

(b) レコード識別子設定段階の動作例 (図1ステップS14)

既読レコード数変数 $rr \leftarrow [rr] + 1$  (インクリメントする。)  
 レコード識別子設定テーブル $rid-t$  にレコード識別子 $rid \leftarrow [rr]$ 、開始アドレスオフセット値、レコード長をセットする。  
 レコード順ソートテーブル $r-st$ にレコード識別子 $rid \leftarrow [rr]$ を登録す

(c) キー欠落処理段階の動作例 (図1ステップS110)

レコード順ソートテーブル $r-st$ の当該レコード識別子 $rid$  行に当該キー欠落フラグを立てる。  
 第 $k$  優先キー初期状態遷移表 $tid-k(0)$ の最終遷移状態フラグ欄を1加算し、受理レコード識別子リスト欄に当該レコード識別子 $rid$  をアペンドする。

(d) レコード識別子設定テーブル $rid-t$  の構造例

レコード識別子 $rid$	開始アドレス オフセット値	レコード長
1	0	45
2	45	40
3	85	45
4	130	38
5	168	38
6	.	.
.	.	.

【図 6】

## ソートテーブルのデータ構造例

(a) 順序値タプルj-tp1、複数順序値タプルのデータ構造例

レコード識別子rid	第1優先キー 順序値jval(1)	第2優先キー 順序値jval(2)	...	第K優先キー 順序値jval(K)
------------	----------------------	----------------------	-----	----------------------

(b) ソート順ソートテーブルs-stの構造例

順序値jval	レコード識別子rid
1	3 0 1
2	1 5 8
3	2 3
3	1 6 8 7
5	1 4
.	.

注) 一般に順序値jval  
は同一値を複数レコード  
識別子ridに対応させる  
ことができる。

(c) レコード順ソートテーブルr-stの構造例

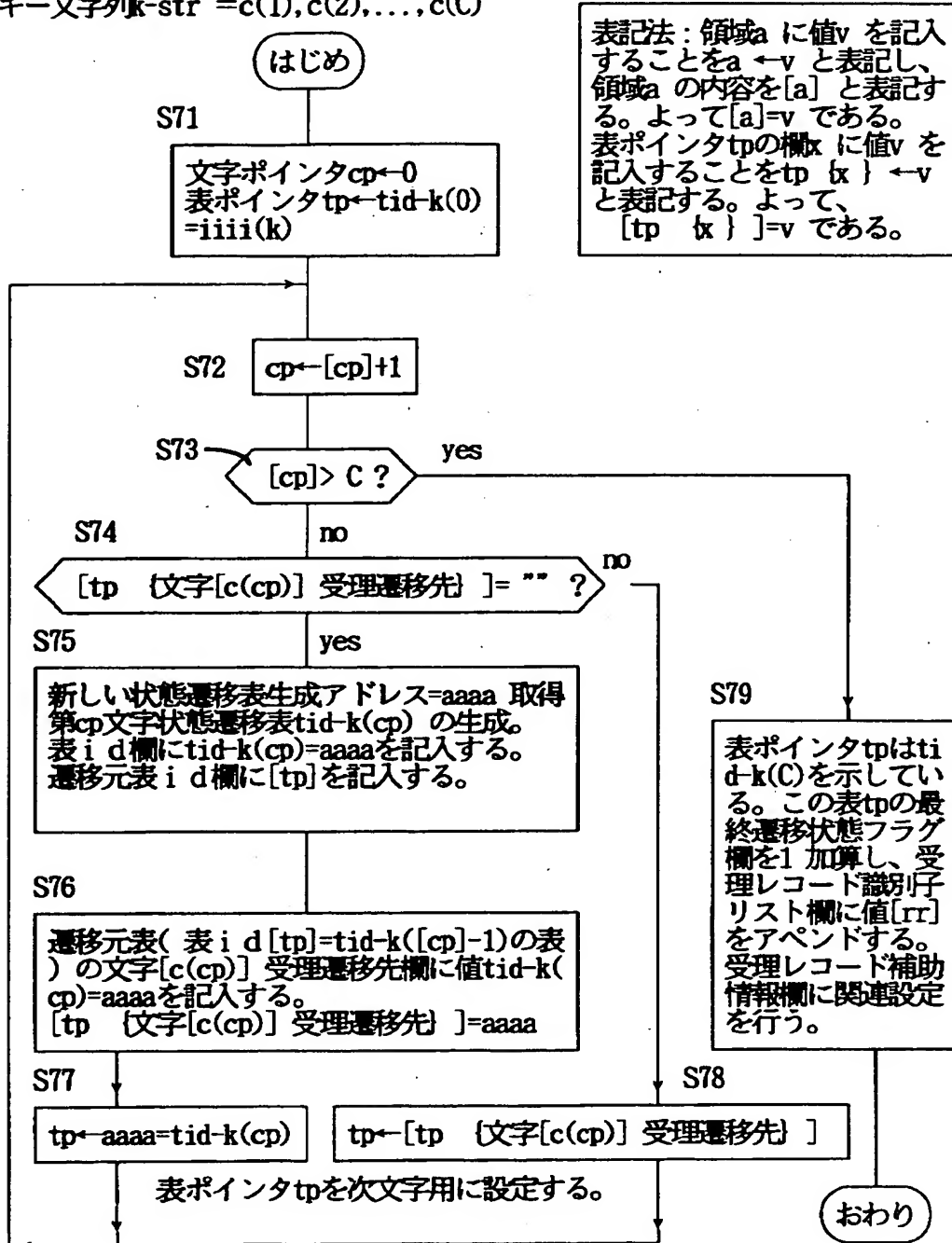
レコード 識別子rid	第1優先キー		第2優先キー		第K優先キー	
	キー 欠落 フラグ	順序値 jval	キー 欠落 フラグ	順序値 jval	キー 欠落 フラグ	順序値 jval
1		251		68		106
2		38		497		184
3	欠落	max(1)		711		992
4		574		25		78
5		398		56	欠落	max(K)
6		16	欠落	max(2)		532
.	.	.	.	.	.	.

注) 順序値タプル生成段階においてmax(1), max(2), ..., max(K)の値が決定する。

【図 7】

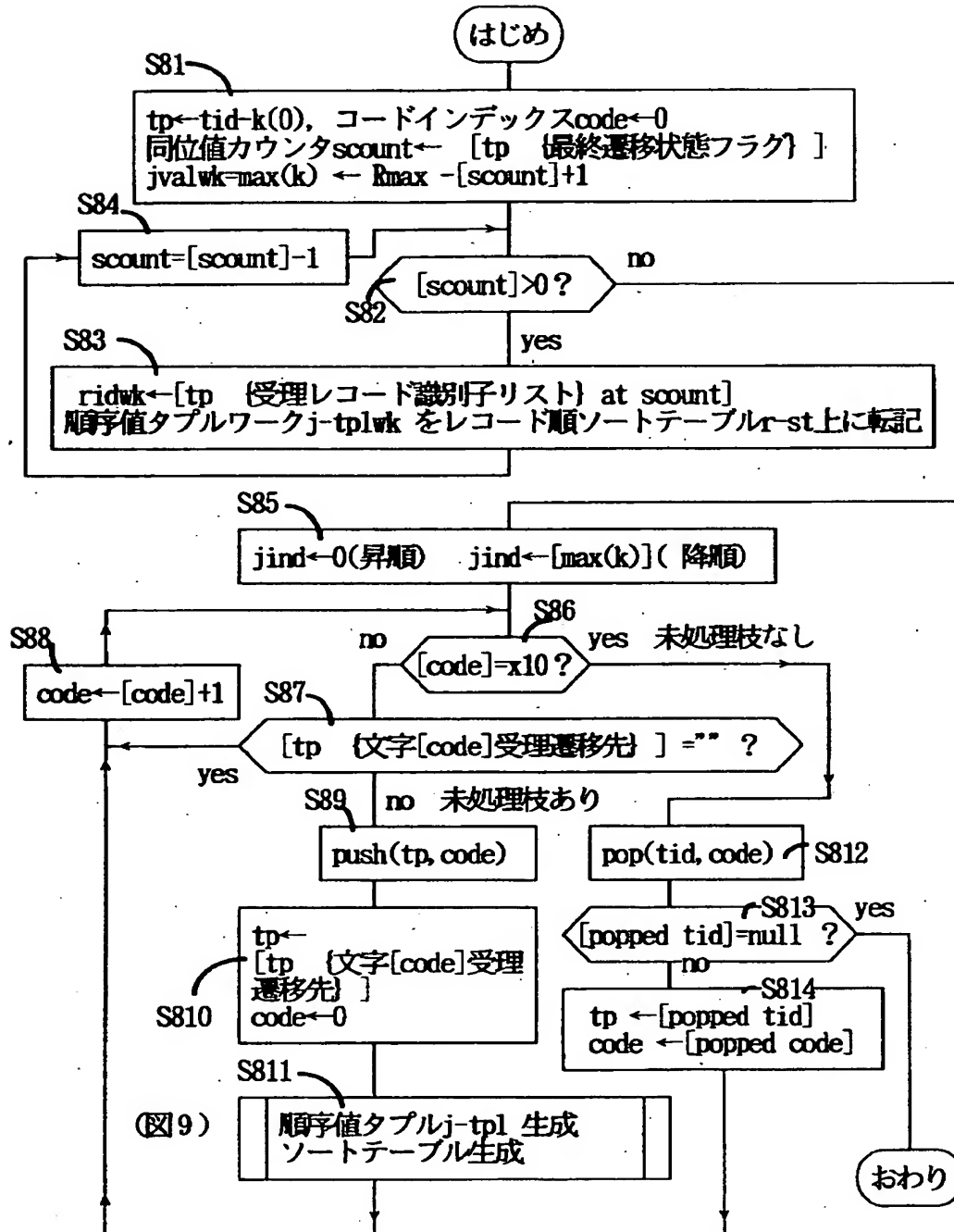
## オートマトン作成処理段階の動作フロー例

キー文字列k-str = c(1), c(2), ..., c(C)



【図 8】

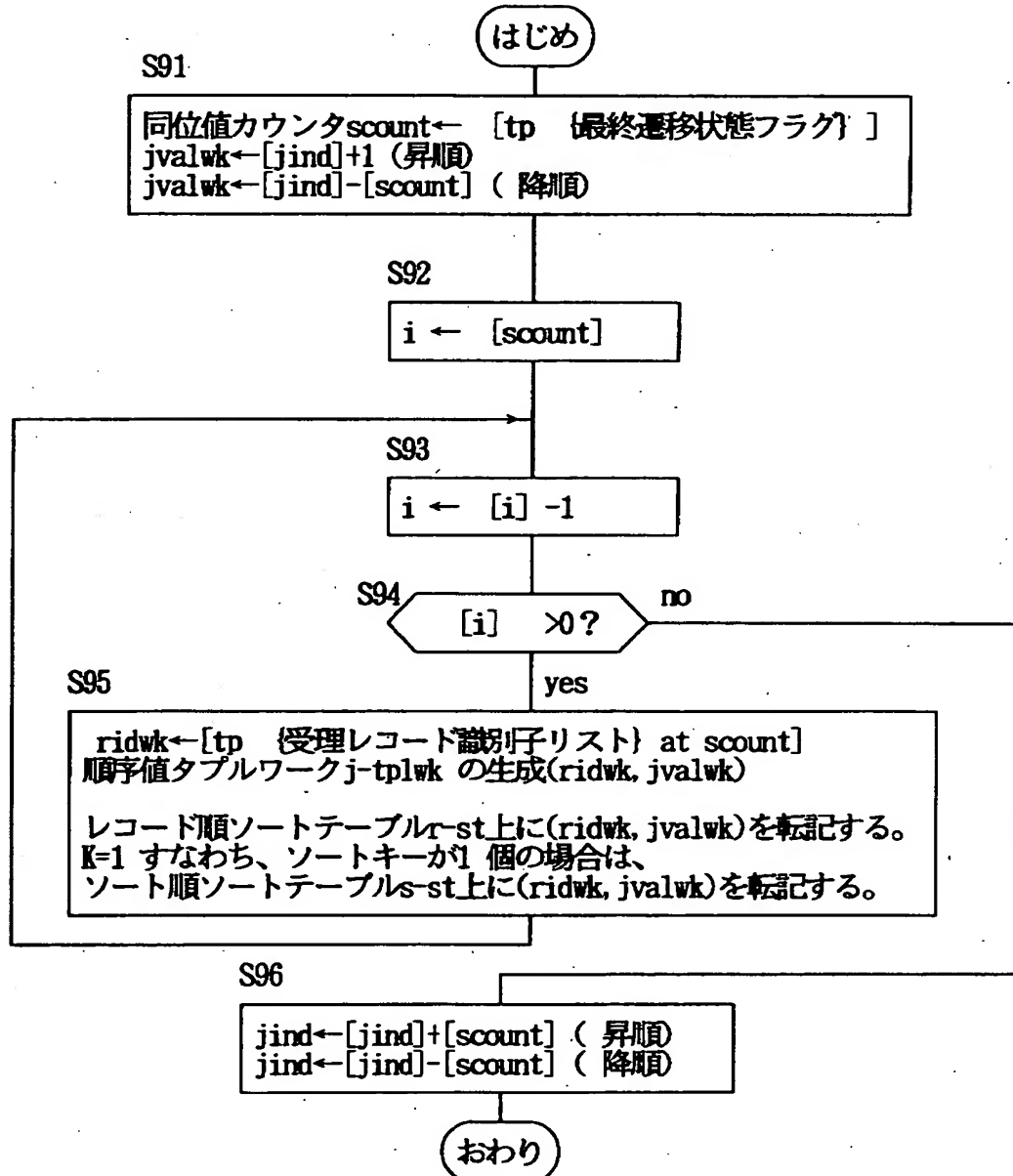
ソートテーブル作成処理段階の動作フロー例



【図 9】

## 順序値タプルj-tp1 生成・ソートテーブル生成ルーチンの動作例

入力： 順序値インデックスjind、ソートモードフラグsm(k)、表ポインタtp  
 出力： 順序値タプルj-tp1 (順序値タプルワークj-tp1wk)  
 レコード順ソートテーブルr-st、ソート順ソートテーブルs-st(オプション)  
 順序値インデックスjind



【書類名】        要約書

【要約】

【課題】    指定されたソートキー項目に従ってレコードを昇順または降順に並べ替えるデータソート方法で、従来のデータソート方法より高速なデータソート方法を提供する。

【解決手段】    各レコードのソートキー項目の文字列を受理するオートマトンで、かつその最終遷移状態に当該レコード識別子を対応付けたオートマトンを生成するオートマトン作成処理段階と、前記オートマトンを走査して前記レコードを昇順または降順に順番付けた順序値と当該レコード識別子との組である順序値タプルを生成する順序値タプル生成段階と、前記順序値タプルを所定の順に並べたソートテーブルを生成するソートテーブル作成処理段階と、を有するデータソート方法によって、従来のデータソート方法より高速なデータソートが実現する。

【選択図】        図1

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社